2024

Query a Digital Music Store Database

USING SQL FOR DATA ANALYSIS PROJECT

SEÇKİN YILMAZ

Contents

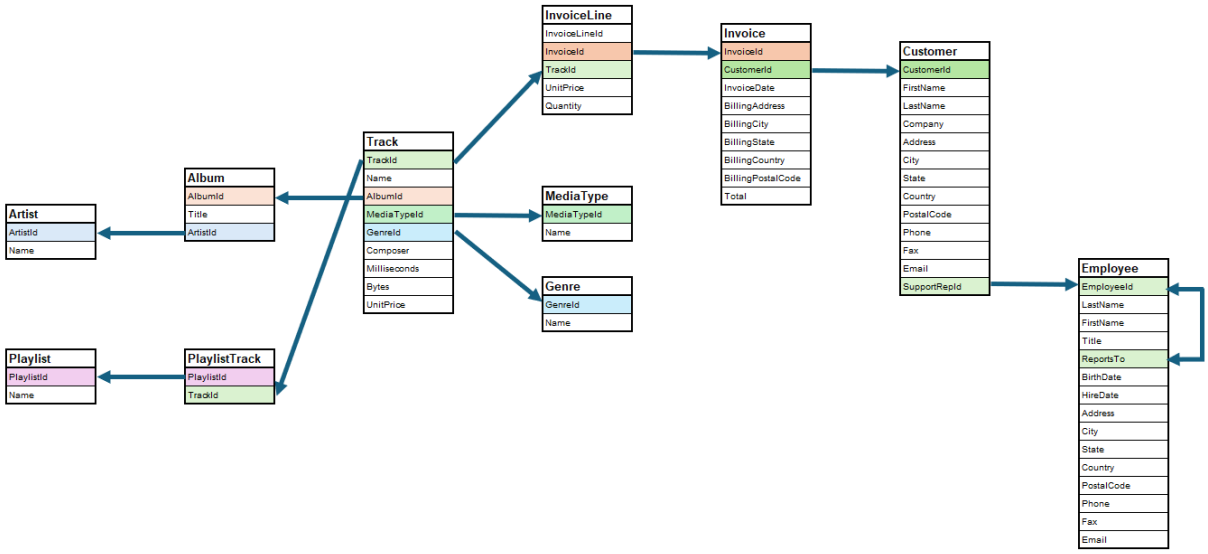
Project

Step 1: Get the Chinook database up and running.

Instructions

You can run the **DB Browser** database on your own machine (See the ***Set Up the DB Browser Database*** page) or you can use the provided workspace (see the ***Chinook Postgres SQL Workspace*** page)

Chinook Database & Relationships



Setup Database Browser

SQLite is used for this project. Database downloaded from GitHub. All setup performed according to instructions.

Check the Setup

To check the database setup following queries are performed and checked the results.

Query	Result
`SELECT count(Name) FROM Track WHERE Composer = 'U2';`	44
`SELECT max(Total) FROM Invoice WHERE BillingCountry = 'Spain';`	13.86
`SELECT Title FROM Employee WHERE LastName = 'Johnson';`	Sales Support Agent

Step 2: Complete the SQL Question Sets

Instructions

This series of questions will confirm that you have mastered the main concepts taught throughout the SQL lessons. **These questions will not be "graded" by a reviewer and are intended to help you self-assess.**

Question Set – 1

Question – 1: ***Which countries have the most Invoices?***

Use the **Invoice** table to determine the countries that have the most invoices. Provide a table of **BillingCountry** and **Invoices** ordered by the number of invoices for each country. The country with the most invoices should appear first.

Solution & Query:

SELECT BillingCountry, count(InvoiceId) as Invoices

FROM Invoice

GROUP BY BillingCountry

ORDER BY 2 DESC

Result

	BillingCountry	Invoices
1	USA	91
2	Canada	56
3	France	35
4	Brazil	35
5	Germany	28
6	United Kingdom	21

Question – 2: ***Which city has the best customers?***

We want to throw a promotional Music Festival in the city we made the most money. Write a query that returns the 1 city that has the highest sum of invoice totals. Return both the city name and the sum of all invoice totals.

Solution & Query:

SELECT BillingCity, sum(total) as sum_of_total_invoice

FROM Invoice

GROUP BY 1

ORDER BY 2 DESC

LIMIT 1

Result

	BillingCity	sum_of_total_invoice
1	Prague	90.24

Question – 3: ***Who is the best customer?***

The customer who has spent the most money will be declared the best customer. Build a query that returns the person who has spent the most money. I found the solution by linking the following three: **Invoice**, **InvoiceLine**, and **Customer** tables to retrieve this information, but you can probably do it with fewer!

Solution & Query:

SELECT

CONCAT(Customer.FirstName, ' ', Customer.LastName) as Customer_Name, sum(Total) as

Total_Spend_of_Customer

FROM Invoice

JOIN Customer

```
ON Customer.CustomerId = Invoice.CustomerId

GROUP By 1

ORDER By 2 DESC

LIMIT 1
```

Result

	Customer_Name	Total_Spend_of_Customer
1	Helena Holý	49.62

Question Set – 2:

Question – 1:

Use your query to return the email, first name, last name, and Genre of all **Rock** Music listeners (Rock & Roll would be considered a different category for this exercise). Return your list ordered alphabetically by email address starting with A.

I chose to link information from the **Customer**, **Invoice**, **InvoiceLine**, **Track**, and **Genre** tables, but you may be able to find another way to get at the information.

Solution & Query:

```
SELECT

Customer.Email, Customer.FirstName, Customer.LastName, Genre.Name AS Genre

FROM Customer

INNER JOIN Invoice

ON Invoice.CustomerId = Customer.CustomerId

INNER JOIN InvoiceLine

ON InvoiceLine.InvoiceId = Invoice.InvoiceId

INNER JOIN Track

ON Track.TrackId = InvoiceLine.TrackId

INNER JOIN Genre

ON Genre.GenreId = Track.GenreId

WHERE Genre = 'Rock'

GROUP BY 1,2,3,4

ORDER BY 1;
```

Result

	Email	FirstName	LastName	Genre
1	aaronmitchell@yahoo.ca	Aaron	Mitchell	Rock
2	alero@uol.com.br	Alexandre	Rocha	Rock
3	astrid.gruber@apple.at	Astrid	Gruber	Rock
4	bjorn.hansen@yahoo.no	Bjørn	Hansen	Rock
5	camille.bernard@yahoo.fr	Camille	Bernard	Rock
6	daan_peeters@apple.be	Daan	Peeters	Rock

Question – 2: ***Who is writing the rock music?***

Now that we know that our customers love rock music, we can decide which musicians to invite to play at the concert.

Let's invite the artists who have written the most rock music in our dataset. Write a query that returns the **Artist** name and total track count of the top 10 rock bands.

You will need to use the **Genre**, **Track** , **Album**, and **Artist** tables.

Solution & Query:

```
SELECT

Artist.ArtistId,

Artist.Name As Name,

COUNT(Genre.name) AS Songs

FROM Artist

INNER JOIN Album

ON Album.ArtistId = Artist.ArtistId

INNER JOIN Track

ON Track.AlbumId = Album.AlbumId

INNER JOIN Genre

ON Genre.GenreId = Track.GenreId

WHERE Genre.Name = 'Rock'

GROUP BY 2

ORDER BY 3 DESC

LIMIT 10;
```

Result

	ArtistId	Name	Songs
1	22	Led Zeppelin	114
2	150	U2	112
3	58	Deep Purple	92
4	90	Iron Maiden	81
5	118	Pearl Jam	54
6	152	Van Halen	52

*Question – 3: First, find which artist has earned the most according to the **InvoiceLines**?*

Now use this artist to find which customer spent the most on this artist.

For this query, you will need to use the **Invoice**, **InvoiceLine**, **Track**, **Customer**, **Album**, and **Artist** tables.

Notice, this one is tricky because the **Total** spent in the **Invoice** table might not be on a single product, so you need to use the **InvoiceLine** table to find out how many of each product was purchased, and then multiply this by the price for each artist.

Solution & Query:

a. **The artist has earned the most according to the InvoiceLines:**

```
SELECT Artist.Name, SUM(InvoiceLine.UnitPrice * InvoiceLine.Quantity) as Earned

FROM InvoiceLine

JOIN Track

ON Track.TrackId = InvoiceLine.TrackId

JOIN Album

ON Album.AlbumId = Track.AlbumId

JOIN Artist

ON Artist.ArtistId = Album.ArtistId

GROUP BY 1

ORDER BY 2 DESC
```

Result

	Name	Earned
1	Iron Maiden	138.6
2	U2	105.93
3	Metallica	90.09
4	Led Zeppelin	86.13
5	Lost	81.59
6	The Office	49.75

b. The customer spent the most on this artist:

```
SELECT Artist.Name, sum(InvoiceLine.UnitPrice * InvoiceLine.Quantity) as AmountSpent,
Customer.CustomerId, Customer.FirstName, Customer.LastName

FROM Customer

JOIN Invoice

ON Invoice.CustomerId = Customer.CustomerId

JOIN InvoiceLine

ON InvoiceLine.InvoiceId = Invoice.InvoiceId

JOIN Track

ON Track.TrackId = InvoiceLine.TrackId

JOIN Album

ON Album.AlbumId = Track.AlbumId

JOIN Artist

ON Artist.ArtistId = Album.ArtistId

WHERE Artist.Name = (

        SELECT sub.Most_Earned_Artist

        FROM (

                SELECT Artist.Name as Most_Earned_Artist, SUM(InvoiceLine.UnitPrice *
InvoiceLine.Quantity) as Earned

                FROM InvoiceLine

                JOIN Track
```

```
ON Track.TrackId = InvoiceLine.TrackId
JOIN Album
ON Album.AlbumId = Track.AlbumId
JOIN Artist
ON Artist.ArtistId = Album.ArtistId
GROUP BY 1
ORDER BY 2 DESC
LIMIT 1
) sub)
GROUP BY 4
ORDER BY 2 DESC
```

Result

	Name	AmountSpent	CustomerId	FirstName	LastName
1	Iron Maiden	17.82	55	Mark	Taylor
2	Iron Maiden	15.84	35	Madalena	Sampaio
3	Iron Maiden	13.86	36	Hannah	Schneider
4	Iron Maiden	13.86	16	Frank	Harris
5	Iron Maiden	8.91	27	Patrick	Gray
6	Iron Maiden	8.91	5	František	Wichterlová

Question Set – 3:

Question – 1:

We want to find out the most popular music Genre for each country. We determine the most popular genre as the genre with the highest amount of purchases. Write a query that returns each country along with the top Genre. For countries where the maximum number of purchases is shared, return all Genres.

You will need to use the **Invoice**, **InvoiceLine**, **Customer**, **Track**, and **Genre** tables for this query.

Solution & Query:

a. Find to Invoice.BillingCountry is equal to Customer.Country

```
SELECT COUNT(*) AS Different_Count
```

```
FROM (
  SELECT I.BillingCountry,
      C.Country,
      CASE
        WHEN I.BillingCountry = C.Country THEN 'Same'
        ELSE 'Different'
      END AS Country_Match
  FROM Invoice I
  JOIN Customer C ON C.CustomerId = I.CustomerId
) AS CountryComparison
WHERE Country_Match = 'Different';
```

Result:

	Different_Count
1	0

➔ Therefore I can use the **Invoice.BillingCountry** as **Customer's Country**

b. Create a query to find purchases for each country and genre.

```
SELECT Invoice.BillingCountry as Country_Name, Genre.Name as Genre_Name, count(*) as
Purchases, Genre.GenreId
FROM Invoice
JOIN InvoiceLine
ON Invoice.InvoiceId = InvoiceLine.InvoiceId
JOIN Track
ON Track.TrackId = InvoiceLine.TrackId
JOIN Genre
ON Track.GenreId = Genre.GenreId
GROUP BY 1,2
ORDER BY 1
```

	Country_Name	Genre_Name	Purchases	GenreId
1	Argentina	Alternative & Punk	9	4
2	Argentina	Easy Listening	2	12
3	Argentina	Jazz	2	2
4	Argentina	Latin	8	7
5	Argentina	Metal	7	3
6	Argentina	Rock	9	1

c. Now, select the most purchases from each country

```
SELECT
  sub.Country_Name,
  sub.Genre_Name,
  sub.Purchases,
  sub.Genre_ID
FROM (
  SELECT
    Invoice.BillingCountry AS Country_Name,
    Genre.Name AS Genre_Name,
    COUNT(*) AS Purchases,
    Genre.GenreId AS Genre_ID,
    ROW_NUMBER() OVER (PARTITION BY Invoice.BillingCountry ORDER BY COUNT(*) DESC) AS
Row_Num
  FROM Invoice
  JOIN InvoiceLine
    ON Invoice.InvoiceId = InvoiceLine.InvoiceId
  JOIN Track
    ON Track.TrackId = InvoiceLine.TrackId
  JOIN Genre
    ON Track.GenreId = Genre.GenreId
  GROUP BY 1, 2, Genre.GenreId
) sub
WHERE sub.Row_Num = 1
ORDER BY sub.Country_Name;
```

	Country_Name	Genre_Name	Purchases	Genre_ID
1	Argentina	Alternative & Punk	9	4
2	Australia	Rock	22	1
3	Austria	Rock	15	1
4	Belgium	Rock	21	1
5	Brazil	Rock	81	1
6	Canada	Rock	107	1

The result is wrong. Because some countries have same purchases for different genres. Therefore, query should change as.

```
WITH Country_Genre_Purchases AS (
  SELECT
    Invoice.BillingCountry AS Country_Name,
    Genre.Name AS Genre_Name,
    COUNT(*) AS Purchases
  FROM Invoice
  JOIN InvoiceLine
    ON Invoice.InvoiceId = InvoiceLine.InvoiceId
  JOIN Track
    ON Track.TrackId = InvoiceLine.TrackId
  JOIN Genre
    ON Track.GenreId = Genre.GenreId
  GROUP BY 1, 2
)
SELECT
  Country_Name,
  Genre_Name,
  Purchases
FROM Country_Genre_Purchases AS cgp
WHERE Purchases = (
```

```
    SELECT MAX(Purchases)
    FROM Country_Genre_Purchases AS sub
    WHERE sub.Country_Name = cgp.Country_Name
)
ORDER BY Country_Name;
```

	Country_Name	Genre_Name	Purchases
1	Argentina	Alternative & Punk	9
2	Argentina	Rock	9
3	Australia	Rock	22
4	Austria	Rock	15
5	Belgium	Rock	21
6	Brazil	Rock	81

Question – 2:

Return all the track names that have a song length longer than the average song length. Though you could perform this with two queries. Imagine you wanted your query to update based on when new data is put in the database. Therefore, you do not want to hard code the average into your query. You only need the **Track** table to complete this query.

Return the **Name** and **Milliseconds** for each track. Order by the song length with the longest songs listed first.

Solution & Query:

a. Firstly, find the average song length

```
SELECT ROUND(AVG(Milliseconds),2) as Avg_Track_Length
FROM Track
```

	Avg_Track_Length
1	393599.21

b. Then write the final query

```
SELECT Name, ROUND(Milliseconds, 2) as Track_Lengt
FROM Track
WHERE Track_Lengt > (SELECT ROUND(AVG(Milliseconds),2) AS Avg_Track_Length FROM Track)
ORDER BY 2 DESC
```

Result

	Name	Track_Lengt
1	Occupation / Precipice	5286953.0
2	Through a Looking Glass	5088838.0
3	Greetings from Earth, Pt. 1	2960293.0
4	The Man With Nine Lives	2956998.0
5	Battlestar Galactica, Pt. 2	2956081.0
6	Battlestar Galactica, Pt. 1	2952702.0

Question – 3:

Write a query that determines the customer that has spent the most on music for each country. Write a query that returns the country along with the top customer and how much they spent. For countries where the top amount spent is shared, provide all customers who spent this amount.

You should only need to use the **Customer** and **Invoice** tables.

Solution & Query:

a. Join the both Customer and Invoice then write the query

SELECT Customer.Country, sum(Invoice.Total) as TotalSpent, Customer.FirstName, Customer.LastName, Customer.CustomerId

FROM Customer

JOIN Invoice

ON Invoice.CustomerId = Customer.CustomerId

GROUP BY Customer.Country

Result

	Country	sum(Invoice.Total)	FirstName	LastName	CustomerId
15	Ireland	45.62	Hugh	O'Reilly	46
16	Italy	37.62	Lucas	Mancini	47
17	Netherlands	40.62	Johannes	Van der Berg	48
18	Norway	39.62	Bjørn	Hansen	4
19	Poland	37.62	Stanisław	Wójcik	49
20	Portugal	77.24	João	Fernandes	34

Step 3: Create the Presentation

Instructions

There isn't a 'right answer' for this portion of the project. The project is designed to allow you to be creative in the questions you ask and then write a SQL query to pull the data needed to answer your question successfully.

You will use the pulled data to build a visual (bar chart, histogram, or another plot) that answers your question and present that information in a slide deck using PowerPoint, Google Slides or your favorite presentation software. See more information about what is required and how to submit the project on the ***Project Submission Details*** page.

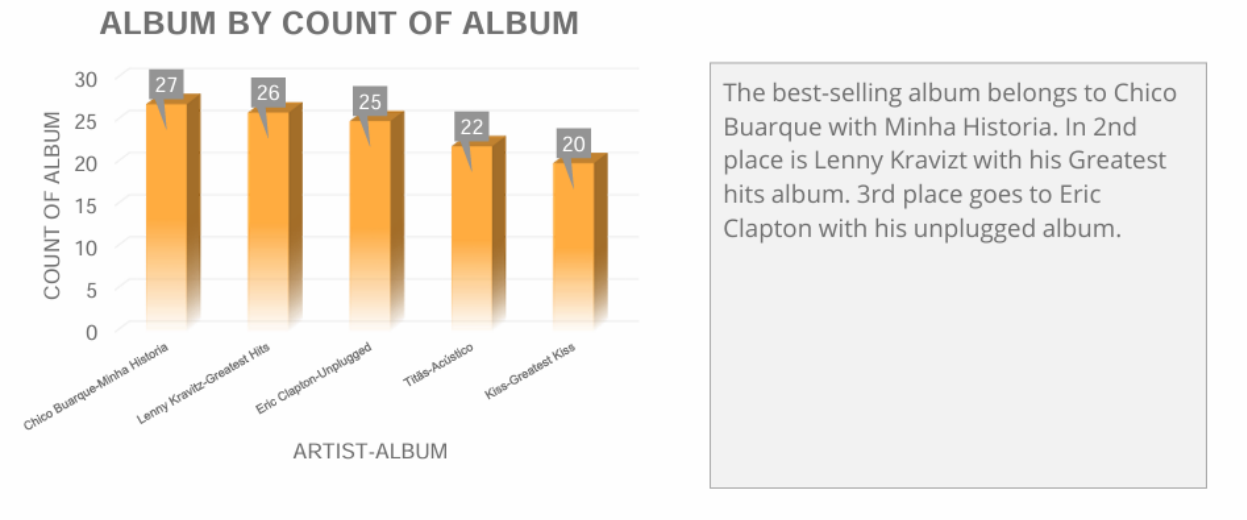
/* Top 5 Artists with the Most Albums */

```
SELECT Artist.Name, count(AlbumId)
FROM Album
JOIN Artist
ON Artist.ArtistId = Album.ArtistId
GROUP BY 1
ORDER BY 2 DESC
LIMIT 5
```

/* Top 5 best-selling albums */

```
SELECT CONCAT(Artist.Name, '-', Album.Title) as Artis_Album, sum(InvoiceLine.Quantity) as Total_Sale_Qty

FROM InvoiceLine

JOIN Track

ON Track.TrackId = InvoiceLine.TrackId

JOIN Album

ON Album.AlbumId = Track.AlbumId

JOIN Artist

ON Artist.ArtistId = Album.ArtistId

GROUP BY 1

ORDER BY 2 DESC

LIMIT 5
```

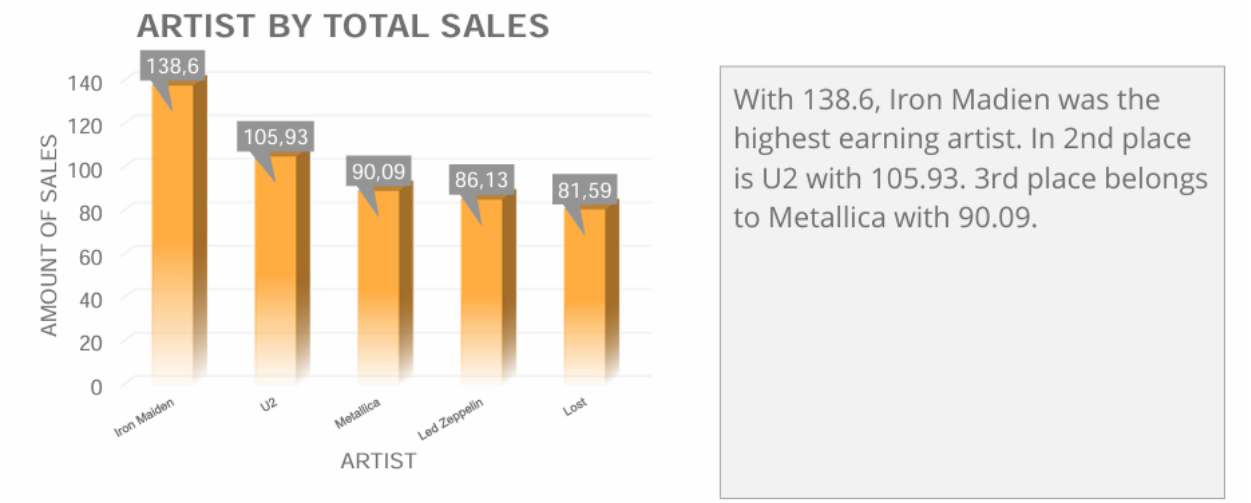
Top 5 best-selling albums




```
/*Top fiveartists with the highest earnings */

SELECT

  ar.ArtistId,

  ar.Name AS ArtistName,

  SUM(il.UnitPrice * il.Quantity) AS TotalEarnings

FROM
```

```
    Artist ar
JOIN
    Album al ON ar.ArtistId = al.ArtistId
JOIN
    Track t ON al.AlbumId = t.AlbumId
JOIN
    InvoiceLine il ON t.TrackId = il.TrackId
GROUP BY
    ar.ArtistId, ar.Name
ORDER BY
    TotalEarnings DESC
LIMIT 5;
```

Top five artists with the highest earnings

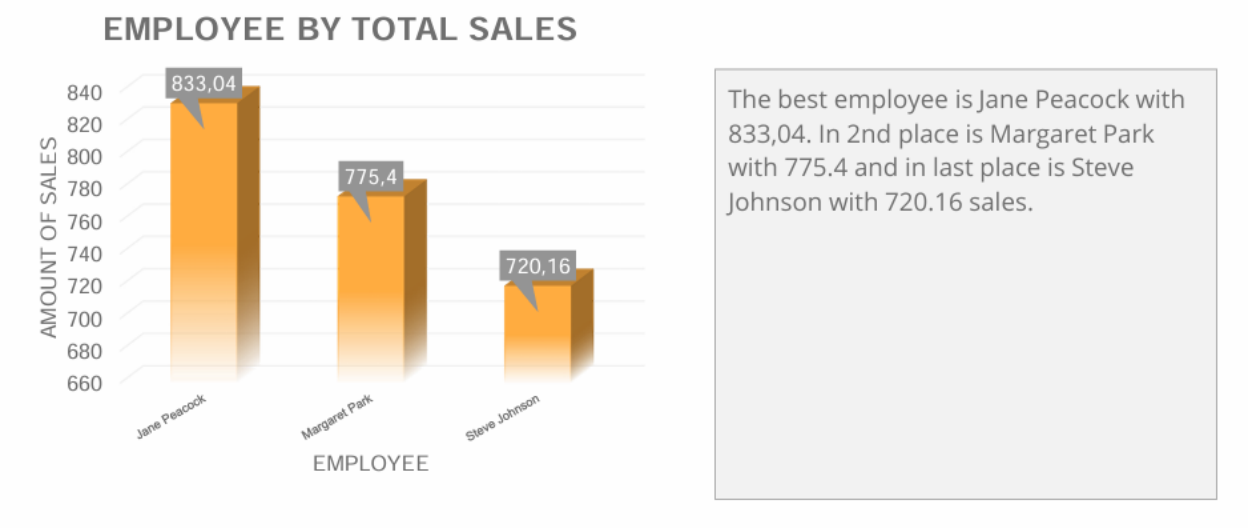


With 138.6, Iron Madien was the highest earning artist. In 2nd place is U2 with 105.93. 3rd place belongs to Metallica with 90.09.

```
/* What is the total sales amount of each employee? */
SELECT
    Employee.EmployeeId,
    CONCAT(Employee.FirstName, ' ', Employee.LastName) AS EmployeeName,
    SUM(InvoiceLine.UnitPrice * InvoiceLine.Quantity) AS TotalSales
```

```
FROM
   Employee
JOIN
   Customer ON Employee.EmployeeId = Customer.SupportRepId
JOIN
   Invoice ON Customer.CustomerId = Invoice.CustomerId
JOIN
   InvoiceLine ON Invoice.InvoiceId = InvoiceLine.InvoiceId
GROUP BY
   1, 2
ORDER BY
   3 DESC;
```

What is the total sales amount of each employee?



The best employee is Jane Peacock with 833,04. In 2nd place is Margaret Park with 775.4 and in last place is Steve Johnson with 720.16 sales.